Understanding Factors Influencing the Creation of Personal Applications in Knowledge Management

Michael Kohlegger

Veröffentlicht in:
Multikonferenz Wirtschaftsinformatik 2012
Tagungsband der MKWI 2012
Hrsg.: Dirk Christian Mattfeld; Susanne Robra-Bissantz

Braunschweig: Institut für Wirtschaftsinformatik, 2012
Understanding Factors Influencing the Creation of Personal Applications in Knowledge Management

Michael Kohlegger
University of Innsbruck School of Management, Department for Information Systems, Production and Logistics Management, 6020 Innsbruck, Austria,
E-Mail: michael.kohlegger@uibk.ac.at

Abstract
Knowledge management is increasingly getting ‘on-demand’ and ‘just-in-time’. One sign of this paradigm change is the ever increasing emergence of applications that are created by the users themselves to support their personal needs. Despite their opportunities, these applications pose several risks on corporate knowledge management, e.g., the transgression of organisational policies or the profusion of work time. For that reason it is important for corporate knowledge management to understand these self-initiatives. This paper presents the results of a structuring content analysis in end-user development literature whose objective was the investigation of factors influencing the creation of applications by end-users. It presents a typology of factors and discusses their relations and implications.

1 Personal Applications in Knowledge Management
Knowledge – humans’ cognitive expectations, either rational or intentional, that are used to interpret situations or generate behaviour, activities or solutions [5] – is today referred to as one of the key resources of organisational compatibility [37] and as product or property [35]. For that reason many organisations have introduced programs that especially focused on the management of this resource – initiatives on knowledge management (KM) [e.g., 67, 62, 55]. Though KM should not solely rest on technologies, they can help to facilitate processes by reducing costs or imposing control and continuity [68, 8]. With technological innovation, KM has changed and changes strongly. Tsui [66, 11] proposed three major shifts in KM for the future. These are (i) an increased alignment of KM technologies with business process management tools, (ii) a shift of KM technologies towards an ‘on-demand’ or ‘just-in-time’ paradigm and, resulting from this, (iii) an ever increasing emergence of “software applications developed [or] selected by an individual to support his/her daily work tasks […]” (p. 4). These personal applications are technical artefacts which solidify individual knowledge in a de-contextualised form and are used during the conduction of corporate processes. For this reason they can be located on the border between human (personalized) and technology-oriented (codified) KM [5].
Understanding this phenomenon and the opportunities and risks that it imposes is thus subject to either form of KM.

The increasing awareness of personal applications does not exclusively concern KM. Similar concepts, available in literature, are, e.g., *situational applications* [25], *situated software* [18], *end-user applications* [50] or mashed-up applications [69, 70]. Such on demand solutions can help to resolve the paradoxical situation that the large, yet still increasing, number of individuals in professional service work [71, 72, 73] faces today. Many of them are performing knowledge work which is engaged in the work with abstract knowledge, under ill-structured and creative work conditions, often in need of strong formal education and producing rather than merely manipulating knowledge [5, 28, 57, 14]. These persons are in need of flexible personalized IT [5] but at the same time work under conditions where traditional IT is unable to follow their demands. They often face a dynamic and unpredictable environment that “requires the ability to create just-in-time solutions to address unique situations without waiting for the IT department” [25] (p. 17). Furthermore, knowledge workers often have unique understanding of their tasks which is not easily articulated to external developers [25].

Personal applications can provide a solution to this issue. Despite their opportunities, which have already been named before, they impose several challenges on organisations. The creation of on-demand solutions might result in uncontrolled proliferation [49]. Their development model, often described as ‘quick-and-dirty’ [70], may be a reason for lacking documentation [56], quality [49] or maintainability [49]. Additionally, these applications might threaten or directly undermine corporate policies [40]. For that reason it is important for organisational KM to understand such self-initiatives to be able to respond properly.

In order to contribute to this understanding, this paper reports on a literature research which was conducted to find factors that influencing the creation of personal applications by non-professionals. It concentrated on a large body of research that has been reported on as end-user computing and development. After a short introduction in section one, section two explains how a structuring content analysis can be utilized to approach the work’s research question and how the actual research procedure looked like. In section three, the results of the analysis are given. Finally, section four gives a summary of the results and an outlook on future research.

2 Finding Factors through Structuring Content Analysis

Central objective of this research is to investigate textual sources – i.e., existing literature – for factors that influence personal application development. Literature, commonly, comes up with a large number of explicit findings that can be easily deduced but at the same time includes several findings which are reported between the lines. For that reason neither a solely deductive nor an inductive approach was appropriate. Instead, a combined approach was chosen to answer the inquiry’s central question - Structuring Content Analysis (SCA) [6].

SCA works with research approach that uses an initial set of concepts – the initial category system – which can be extended during the analysis [6]. The benefit of this approach lies in the researcher’s ability to use his/her previous knowledge at the beginning of the analysis while being still able to integrate new findings that emerge during the analysis. These findings are induced by the researcher him/herself. In this respect SCA takes the epistemological position of non-positivism which means that “facts and values are intertwined and hard to disentangle […]
in scientific knowledge” [12] (p. 273). SCA uses a defined action plan (figure 1) for the analysis process [6]. The remaining section explains this procedure.

**Figure 1**: Procedure of a SCA according to Mayring (p. 84)

**Step 1: Identification of analysis content.** Mayring’s SCA primarily relies on textual source material. For that reason, the action plan starts with the identification of relevant sources of information which could be used to answer the inquiry's central question. In the study of factors affecting end-user development of applications it was chosen to concentrate on articles on end-user computing and development that have been published in scientific journals with double blind review. The end-user computing and development domain was chosen as the topic of personal applications in KM shows many similarities to this well covered body of research that, for this reason, may come up with a broad spectrum of insights. The focus had been wittingly on scientific journals as they are supposed to reflect the state of research within a domain and a respective time period [22]. For that reason there is no attempt being made to provide bibliographic coverage of all possible sources of information, in particular books, book section, conference proceedings, dissertations or unpublished work. All articles had to be published in English language and had to be available to the author. The articles were identified through a four step process. This process was partially adopted from [23] and [47] and will be explained below.

Initially, a visual search in the last ten-year issues of the extended AIS Senior Basket of Journals\(^1\) as well as the Journal of Organizational and End User Computing\(^2\) was performed to find all recent articles on end-user computing and development. Then, a Boolean key-word search for articles related to these topics was conducted. Therefore, the search engines ISI Web of Knowledge\(^3\) and Google Scholar\(^4\) were used. For this step, no limitation regarding the articles' publication dates was set. As the returned number of articles produced by Google Scholar was very high but ranked, only the first 200 results were considered. The search used various combinations and forms of the search terms: end-user computing, end-user development, end-user software engineering and end-user application. After having conducted the first two steps, several potentially interesting articles on end-user development have been revealed. In step number three, the bibliographies of these articles were searched for new articles, not yet on the analysis list. Finally, a revers search was performed to identify all articles which cited the identified articles.

Before starting the analysis, all articles where checked for their alignment with the study. Some articles did not include the concept of development into their consideration of end-user computing. They rather viewed end-user computing as the pure use of software and hardware

---


by end-users [e.g., 33, 41]. These articles were cleared from the list. A relatively large number of articles did not focus on issues of describing or managing end-user development but investigated issues of end-user satisfaction [e.g., 32, 38] or end-user training [e.g., 43, 27]. These articles were also excluded from the list. Some articles were too general in focus and did not contain factors influencing end-user development [e.g., 64, 63]. These articles were excluded too. The last group of articles which was excluded was that of articles having a focus which was too narrow, thus maybe giving a biased image. These articles often concentrated on a specific tool for end-user development, e.g., spread sheet software [e.g., 36, 34]. The final list, presented in table 1, included 53 articles from 18 journals.

Table 1: Identified Sources in End-User Computing Literature

<table>
<thead>
<tr>
<th>Source</th>
<th>Perspective</th>
<th>Methodology</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alavi et al. [Ex1]</td>
<td>ORG</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Alavi et al. [Ex3]</td>
<td>ORG</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Alavi et al. [Ex5]</td>
<td>ORG</td>
<td>QUT</td>
<td>31</td>
</tr>
<tr>
<td>Amoroso [Ex7]</td>
<td>ORG</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Amoroso et al. [Ex9]</td>
<td>COM</td>
<td>QUT</td>
<td>508</td>
</tr>
<tr>
<td>Amoroso et al. [Ex11]</td>
<td>EUS</td>
<td>QUT</td>
<td>40</td>
</tr>
<tr>
<td>Beheshtian et al. [Ex13]</td>
<td>ORG</td>
<td>QUL</td>
<td>5</td>
</tr>
<tr>
<td>Benjamin [Ex15]</td>
<td>ORG</td>
<td>QUL</td>
<td>1</td>
</tr>
<tr>
<td>Benson et al. [Ex17]</td>
<td>EUS</td>
<td>QUL</td>
<td>67</td>
</tr>
<tr>
<td>Bergeron et al. [Ex19]</td>
<td>ORG</td>
<td>QUT</td>
<td>212</td>
</tr>
<tr>
<td>Berrisford et al. [Ex21]</td>
<td>ORG</td>
<td>QUL</td>
<td>5</td>
</tr>
<tr>
<td>Bill et al. [Ex23]</td>
<td>EUS</td>
<td>QUT</td>
<td>505</td>
</tr>
<tr>
<td>Brancheau et al. [Ex25]</td>
<td>COM</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Brown et al. [Ex27]</td>
<td>ORG</td>
<td>QUL</td>
<td>5</td>
</tr>
<tr>
<td>Burnett et al. [Ex29]</td>
<td>EUS</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Cheney et al. [Ex31]</td>
<td>ORG</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Cheribakov et al. [Ex33]</td>
<td>EUS</td>
<td>QUL</td>
<td>790</td>
</tr>
<tr>
<td>Choo et al. [Ex35]</td>
<td>ORG</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Cotteker et al. [Ex37]</td>
<td>EUS</td>
<td>QUL</td>
<td>14</td>
</tr>
<tr>
<td>Couger [Ex39]</td>
<td>EUS</td>
<td>QUT</td>
<td>618</td>
</tr>
<tr>
<td>Doll et al. [Ex41]</td>
<td>EUS</td>
<td>QUT</td>
<td>5</td>
</tr>
<tr>
<td>Elder et al. [Ex43]</td>
<td>EUS</td>
<td>QUT</td>
<td>10</td>
</tr>
<tr>
<td>El-Dor et al. [Ex45]</td>
<td>ORG</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Galletta et al. [Ex47]</td>
<td>ORG</td>
<td>CON</td>
<td>-</td>
</tr>
<tr>
<td>Gallivan et al. [Ex49]</td>
<td>EUS</td>
<td>QUT</td>
<td>96</td>
</tr>
<tr>
<td>Guimaraes et al. [Ex51]</td>
<td>ORG</td>
<td>QUT</td>
<td>173</td>
</tr>
<tr>
<td>Hackathorn et al. [Ex53]</td>
<td>EUS</td>
<td>QUT</td>
<td>239</td>
</tr>
</tbody>
</table>

Table 1: Identified Sources in End-User Computing Literature

All articles were categorized according to their scope into focusing on organisational (ORG) or individual aspects of end-user development (EUS). Some articles combined both positions (COM). With respect of work type, articles were found to be either conceptual (CON) or empirical. Empirical work could either be qualitative (QU) or quantitative (QUT).

Step 2: Definition of the structuring dimensions. There are two main streams in end-user computing and development literature. One concentrates on individual aspects of end-user computing and development, e.g., types of end-users [61, 26], utilization [17, 59] or performance [30, 54]. The other stream reports on organizational aspects of end-user computing and development, e.g., risks [15, 24], strategies [16, 20] or support [48, 19].

According to these two streams, the basic structuring dimensions were defined as organisational and individual. All revealed factors influencing personal application development were assigned to one of these two basic dimensions.

Step 3: Definition of category system. The category system is the set of all deduced and induced factors and their relationships with each other. The relationships within this study are solely is a’ relationships. This means that a subordinate node is a concept of a superior node with a higher level of detail. Each node of the category system represents a distinct concept.

---

5 For space restriction reasons, bibliography can be found at http://c437-www.uibk.ac.at/mkwi/list.pdf.
These concepts are aggregations of phenomena which have been found in the textual source material [8]. A phenomenon thereby can be represented by a single word, sentence, paragraph or even by the whole article [8]. In order to work with these concepts they are labelled which is often referred to as coding [7, 8, 9, 10].

Newly discovered concepts were typically classified according to the basic structuring dimensions. Later, when the number of phenomena backing the concept rose and the concept thus became more theoretical saturated it could be rearranged within the typology.

The initial category system was deduced from literature. It consisted of two organisational and one individual factors affecting end-user application development. On the organisational side, organisational control and expansion [53] had been found to be important. On the individual side it was user type [61].

**Step 4: (Pre)definition of coding approach.** As described in the previous section, analysis through coding relies on the abstraction of concepts from phenomena. Therefore, a dictionary of codes is employed which contains rules and examples when a particular code is to be used. This is often referred to as the coding manual [5]. Mayring [6] proposes the use of a coding manual to ensure (i) the comprehensibility of the coding process and (ii) to give the researcher guidance for the conduction of analysis. Each concept should be explicated by the following elements:

- The concept's *definition* specific description of the concepts essential nature. Each definition therefore includes a short form which is used to annotate the text fragments and a long form which contains the whole definition.

- The definition of a respective category is assisted by *examples*. Therefore all already annotated text fragments are associated with the definition. Newly assigned text fragments can thus directly be compared with already assigned ones.

- If neither the definition nor the example fragments can help to delimit categories against each other, additional *coding rules* can be supplied. This is, however, only necessary where differentiation problems exist.

**Step 5/6: Screening of material and extraction of coded fragments.** After having established the initial category system with its coding manual, the actual coding process takes place. This means that the material is read through by the researcher and all relevant parts in the text are assigned to the category system [6]. The actual analysis thereby concentrated on the 'value adding' parts of the article. These are the parts coming up with new findings and interpretations and, thus, can be contrasted from non-value-adding parts where the authors solely concentrate on relating their work to previous research. In this respect my analysis commonly concentrated on the descriptions of methodology, methods and empirical findings as well as the discussion and the article’s conclusion and outlook. Nevertheless, each article was carefully read as a whole piece of work. In order to avoid a bias resulting from the sequence of analysis [2] the articles were selected randomly.

The assignment of codes required that the respective part in the text corresponded to an existing concept. If this was not the case a new concept could be induced. As a consequence of this partially inductive approach that is based on Grounded Theory [8, 3] the category system changed during the analysis. Thus, if the material would only have been worked through once, not all sources could have been coded with all concepts. In literature, there are different views how to deal with this conjuncture. Glaser (p. 442) proposes to start using a code when it occurs
and to re-code only if the code does not become theoretically saturate on the remaining material [39]. As the number of sources in this inquiry was not that large, it was, nevertheless, decided to use three re-coding loops during the analysis [4]. These were done after having completed the first nine, 26 and finally all articles.

**Step 7: Revision of category system.** If a text fragment did not correspond to any category in the category system, but was, nevertheless, deemed valuable in respect of answering the inquiry’s central question, a new concept could be created. The new concept was then assigned to the category system. This meant that it was classified according to the basic structuring dimensions. If the concept could be related to an existing node within the respective dimension it was placed there. Otherwise it was left on the base level.

3 Factors Influencing Personal Application Creation

It has been found that the initially generated structuring dimensions, *individual* and *organisational*, were proper to organize the deduced and induced factors. For that reason these two dimensions were kept as first order nodes.

![A taxonomy of factors potentially influencing end-user self-organisation](image)

Figure 2: A taxonomy of factors potentially influencing end-user self-organisation

On the **individual** level, motivation plays an important role in the creation of personal applications. Motivation has been found to either result from a personal intention to act in a self-initiative way (*intrinsic*) or from an intention which is triggered from outside (*extrinsic*). The latter thereby can be triggered by external stimuli which demand an optional action (*extrinsic pull*) or by such stimuli that demand a mandatory action (*extrinsic push*). Beside motivation, user characteristics have been found to play a role in the creation of personal applications. These have been subsumed under the concept *end-user type*.
On the intrinsic level, attitude plays a key role. As frequently reported, end-users attitude towards the power of end-user development itself (*attitude towards EUD*) as well as their position regarding corporate IT (*attitude towards IT*) are important. Especially when attitude towards end-user development was positive and/or attitude towards corporate IT was negative, end-users were reported to facilitate end-user development. Beheshtian et al. [20] bring this to the point when they say

*When surveyed, many users expressed impatience with the sluggishness of MIS’s response. […] Staff employees felt they were more capable of developing high quality, responsive systems than the MIS department. Nearly all users felt they would develop simple systems that remain a low priority in the MIS backlog.*

There were three extrinsic motivational factors that undoubtedly stand in relation with these attitude concepts. All of which can be found in the above statement. At first, many authors reported that users felt they had a *problem in a niche role*. That meant that they faced a specific situation which they felt was so unique that they did not expect help from corporate IT. This might, to some extent, also be related to the fact that many end-users were described as domain experts whose knowledge would be very difficult to externalise to corporate developers (*knowledge inexpressibility*). So, Berrisford [21], for example, notices that “managers frequently experience difficulty in defining information requirements using the abstractions […] employed by systems analysts” (p. 13). Finally, it was found that end-users were reported to be dissatisfied with provided technical infrastructure (*infrastructure incompatibility*), motivating them to initiate development. Khan [46] for example speaks of “frustration with the centrally provided analytical applications” (p. 286).

There was one group of factors that could be best entitled self-fulfilment. This group contains three concepts that were all subsumed under intrinsic motivation. The first factor within this group was *appetite to learn more*. This concept incorporated the fact that many authors reported that users performed end-user development out of curiosity – in order to learn more about, e.g., a framework. The used code follows the notion of Chambers [24] who reports that “there was a tremendous appetite amongst staff to learn more about Excel”. Another aspect of self-fulfilment was the often reported end-user’s *desires to improve* their environment *proactively and to be self-determined*. Ferneley [13], for example reports of end-users becoming “technology path driver” (p. 181), seeing the “opportunity to change the organisation’s business model” with technology (p. 181) and being “proactive in their use of technology” (p. 179).

End-users desires are, however, limited by the user’s personal attributes. Here, the end-users *bug tolerance* can, e.g., be critical. This means that as personal applications are often reported as ‘quick and dirty’ solutions, end-users need some kind of tolerance against failures in their applications [30, 54, 29]. Another critical user aspect is, of course, the user’s *capability* to perform end-user development. This aspect, however, is sometimes described to be related to the user’s degree of involvement in end-user development. For that reason one could argue that low user capability does not necessarily limit the conduction of development if the user at least wants to develop that application. This, however, might then increase the users *demand for support* which is another aspect of end-user development. Nevertheless, users’ perception of support provision is often reported to be negative. As Mirani et al. [51] report, “[the] support provided to end-users is significantly less than the amount of support needed by them” (p. 160).

While organisational support – in terms of information centres – is always part of some budget,
authors often report of situations where end-users consult third parties, e.g., colleagues who then spent their own work time on the problem – outside of budgets.

This situation was very common as the recourse to support by colleagues was reported very often. Beside simple support, the facilitation of end-user development by the end-user’s community could be found as an important aspect (community facilitation). As reported by Ein-Dor et al. [31] this can result from end-users learning of the possibilities or benefits by watching others or, as described by Cherbakov et al. [25], end-user development becomes beneficial through overcoming a critical mass of users and setting free network effects of the sharing of applications [45].

There have been some work-related factors discovered that influence end-user application development. These are information need, the specific requirements which an end-user’s job imposes on IT and task related motivation, the complexity and volatility of the performed task that poses demands on IT. Whenever IT is not capable of fulfilling end-user’s work requirements, authors report end-users to search for alternatives. Therefore they sometimes even rely on technology which is not provided by their organisation (third-party infrastructure). Beheshtian et al. [20], e.g., describe that “end-users use their home PCs to access data that they may not access at work [...]” (p. 3).

Ein-Dor et al. [31] investigated two factors which they found supported through a quantitative survey. These were job position, where it was shown that “the lower the organizational rank of users, the greater the use of [end-user computing]” (p. 34) and degree of computer use, where it has been found that "output uses are differentially associated with [the] degree of use of [end-user computing]" (p. 34).

Cherbakov et al. [25] raised one final concept which was named level of information on functionality. It was reported that users often start development because of having no information about existing solutions on their problem which may be provided by the organisation or other users. Taylor et al. [65] also highlight this issue when they mention that

“[…] in the organisations studied unless specific mechanisms were established for promulgating awareness of similar end-user application developments, end-user developers appeared to rarely consult each other and simply develop their own systems in isolation […] e.g., in an insurance company, studied as part of this research, a number of underwriters have developed their own spreadsheet-based applications” (p. 90).

What they, however, also mention is that some organisations have been found to undertake some coordination efforts to vanquish this information asymmetry. This has, despite its proximity to the former concept, however, been classified as an organisational factor.

On the organisational level, it was found that many publications concentrated on topics that could be summarised as the organisation’s course of action, employed to reach its end-user computing goals. This concept was labelled strategy. In particular three strategy aspects could be found to influence end-user development: (i) control, (ii) expansion and (iii) evaluation. Control strategy and expansion strategy have been emphasised in many publications as very important to organisational authorities. There has been much research on these topics [49, 44, 52]. Control can be defined as the extent to which an organisation influences intra-organisational end-user development through steering its direction (planning) and limiting its conduction.
Control strategy has been found to be part of a classification framework for organisational strategy regarding end-user computing [53]. It was already incorporated in the initial category system.

*Expansion*, in terms of strategy, is the extent to which the organisation encourages intra-organisational end-user development. Encouragement has thereby been found to be granted through the provision of *education infrastructure, time for development, coordination efforts of similar end-user projects and technical infrastructure*. Especially end-user support has been given a central role as “support is one of the primary mechanisms for managing [end-user computing]” [44] (p. 90) and “organisational support would be positively related to the effectiveness of [end-user computing]” [42] (p. 64).

*Evaluation strategy*, finally, denotes the organisation’s attempt to assess the value of end-user initiatives. Many of the reviewed publications thereby lay emphasis on evaluation by “hard numbers”. Though, this is undoubtedly a good steering instrument for management, a direct evaluation of end-user initiatives is difficult as Rivard [60] quotes a data processing manager

> “Unless we can demonstrate that the applications users develop are profitable to the company, we won’t be able to assert that [a user developed application] is successful. And for the time being, we cannot perform such a demonstration” (p. 44).

An evaluation of end-user initiatives would, however, solve the paradox position in which managers are as they have to fertilize and prune these at the same time and therefore need an indicator for where to put their effort. As Kappelman [44] puts it,

> “Evaluation mechanisms have the potential to provide management, and end-users as well with the information needed to resolve the paradox inherent in simultaneously supporting and controlling, facilitating and restricting, enabling and disciplining, nurturing and regulating, and fostering and restraining EUC” (p. 80).

The demand for evaluation is strongly related to another revealed concept labelled *corporate attitude*. It is the organisation’s predominant position regarding the inter-organisational conduction of end-user development. This concept captions two sub-concepts that were found to have direct influence on the organisation’s behaviour and, via that, indirect influence on end-user development. These were the organisations appraisement of the profitability (*profit*) of end-user development and the *risk* associated with it. Risk of failure of end-user development thereby emerges from a potential miss-delimiting or miss-studying of the target (*analysis risk*), a failure in creating the end-user application (*design risk*) or an unsuccessful employment or operationalization of it (*implementation risk*) [15].

Beside these relatively large constructs that comprise several sub-factors, organisational *size* and *complexity* have been found influencing end-user development. Size matters, as end-user computing in general had been found to be more prevalent in larger firms. This, however, was attested in 1991 and must thus not pertain today. In fact, Raymond [58] himself puts this finding into perspective as “the effect of size is probably indirect through its association with other contextual characteristics”. He names the degree of IT centralisation, the user’s education, his/her information systems sophistication, the organisation’s application portfolio and its systems complexity as some possible contextual mediators. Complexity conceptualises the business conditions that end-users face. The more complex these conditions are, the likelier end-users start to develop their own applications. Raymond [58] describes his findings,
End-user computing was more prevalent in those firms whose information systems had a greater level of organizational complexity [having more users and administrative units per application], confirming hypothesis 10. This is congruent with the notion that [end-user computing] allows for a much greater heterogeneity of individual backgrounds and organizational work units involved in using information resources (p. 209).

4 Summary and Outlook on Future Research

This paper presented the results of a literature review in the end-user computing and development domain. This review was interested in revealing factors that influence the development of personal applications. For this purpose, a SCA was applied.

The present study revealed a typology of factors that can help corporate knowledge management to understand user self-initiatives. This has been shown crucial, as these self-initiatives, in form of personal applications, become increasingly important knowledge management technologies. The scientific contribution of this work lies in the holistic discussion of factors. Though, some of them are very straightforward and have been already explicitly discussed in literature others are not that well supported and need scientific inquiry.

While this study solely concentrated on end-user computing and development literature, other domains should be considered as well. This should include a revision of the found concepts, in particular their wording, definitions and relations, under the use of well-established information systems theories, e.g., task technology fit, technology acceptance model, computer self-efficacy or illusion of control. Another limitation of this study is the temporal distribution of reviewed articles as many of them were published before 1996. For that reason, the found concepts should be confronted with current empirical data, e.g., through investigating knowledge workers self-organisation in real-life settings.

5 Literatur


